Step by step create curd project in laravel

01 - Setup

1. Create or clone project.

```
# -------------- create fresh project ---------------
$ composer create-project laravel/laravel project_name
$ cd project_name

# ------------ clone project -------------------------
$ git clone https://github.com/Nahid-Hassan/job-prep.git
$ ls
job-prep # project
$ cd job-prep
$ ls

nahid@cseru:~/Desktop/blog$ ls
app        bootstrap        composer.lock  database        phpunit.xml  README.md
routes    tests   vite.config.js
artisan  composer.json  config          package.json  public        resources
storage   vendor

$ composer update
```

2. Work with .env file.

```
$ mv .env.example .env
```

3. Goto .env file and change the database name.

```
DB_CONNECTION=mysql
DB_HOST=127.0.0.1
DB_PORT=3306
DB_DATABASE=test
DB_USERNAME=root
DB_PASSWORD=password
```

Then create database inside your `laragon.heidisql -> larvel.mysql`

4. If key is not generate yet...

```
$ php artisan key:generate
```

5. Run your project

```
$ php artisan serve
```

02 - Create migration and then migrate

1. Create migration

```
$ php artisan make:migration create_books_table
```

This command create a migration file named `create_books_table.php` inside `database/migrations/...create_books_table.php`.

Now inside `...create_books_table.php` file,

```php
return new class extends Migration
{
    /**
     * Run the migrations.
     */
    public function up(): void
    {
        Schema::create('books', function (Blueprint $table) {
            $table->id();
            $table->string("title");
            $table->string("author");
            $table->string("isbn");
            $table->integer("price");
            $table->integer("available");
            $table->timestamps();
        });
    }

    /**
     * Reverse the migrations.
     */
    public function down(): void
    {
        Schema::dropIfExists('books');
```

```
        }
};
```

2. Make migrate: create database table inside your database.

```
$ php artisan migrate
```

03 - Database Seeding - (Model, Factory, Seeder)

1. Create Model,

```
$ php artisan make:model Book # Book is the model name
```

This command create a [app/Models/Book.php] file. Now edit Book.php file,

```php
class Book extends Model
{
    use HasFactory;
    protected $fillable = ['title','author','isbn','price','available'];

}
```

2. Create factory,

```
$ php artisan make:factory BookFactory --model=Book
```

Now this command create a database/factories/BookFactory.php] file. Now edit BookFactory.php file.

```php
class BookFactory extends Factory
{
    /**
     * Define the model's default state.
     *
     * @return array<string, mixed>
     */
    public function definition(): array
    {
        return [
            'title' => fake()->name(),
            'author' => fake()->name(),
            'isbn' => fake()->unique()->isbn13(),
            'price' =>fake()->numberBetween(100,300),
```

```
                    'available' =>fake()->numberBetween(0,20)
            ];
        }
    }
```

3. Create database seeder,

```
$ php artisan make:seeder BookSeeder
```

This command create a [database/seeders/BookSeeder.php] file. Now edit this BookSeeder.php
file.

```php
class BookSeeder extends Seeder
{
    /**
     * Run the database seeds.
     */
    public function run(): void
    {
        \App\Models\Book::factory(50)->create();
    }
}
```

and make a changed in database/seeders/DatabaseSeeder.php file.

```php
class DatabaseSeeder extends Seeder
{
    /**
     * Seed the application's database.
     */
    public function run(): void
    {
        // \App\Models\User::factory(10)->create();

        // \App\Models\User::factory()->create([
        //     'name' => 'Test User',
        //     'email' => 'test@example.com',
        // ]);
        $this->call(BookSeeder::class);
    }
}
```
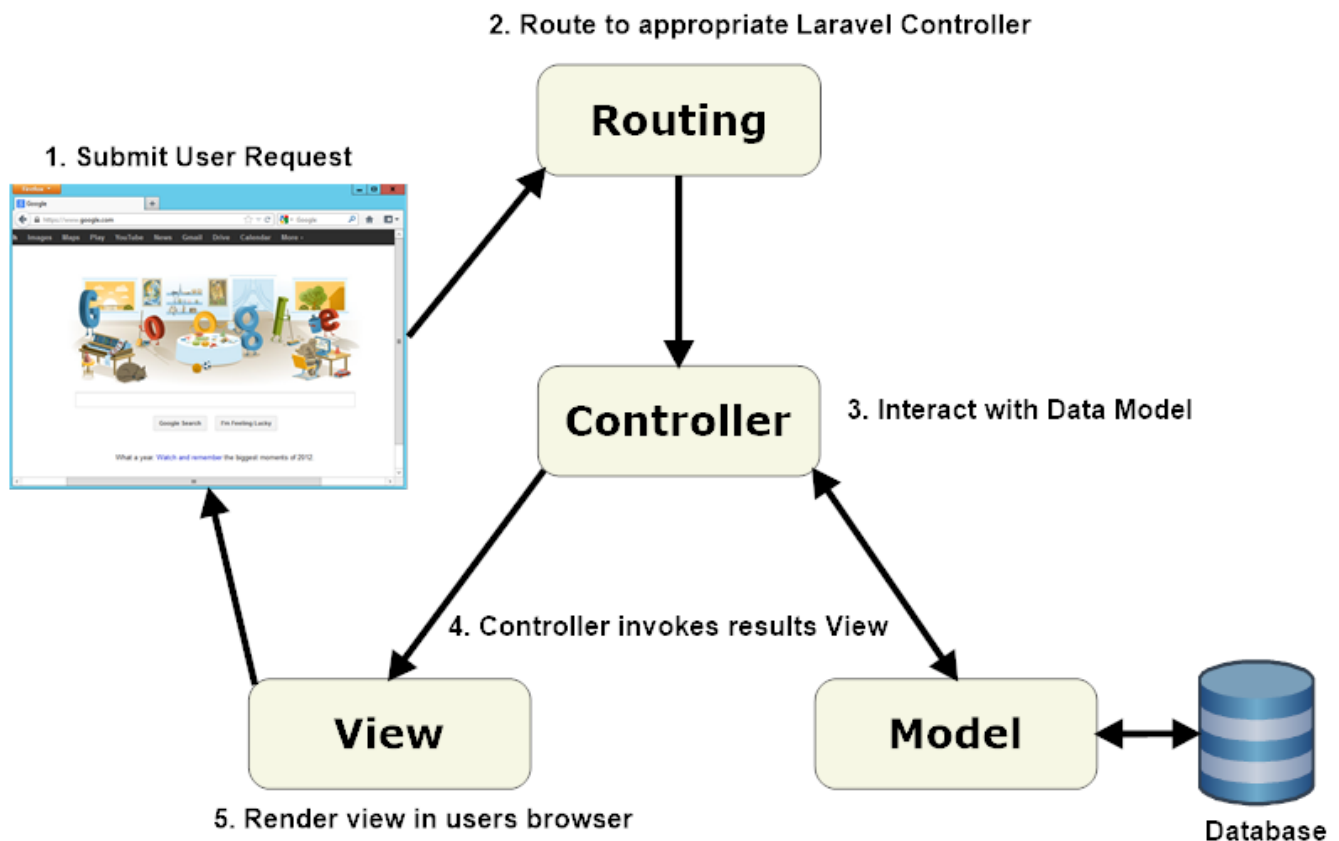
Now run,

```
$ php artisan db:seed
```

Hope successfully seeded data into your database table. **Hurrah...**

04 - Create CRUD Application

1. Create Controller,

```
$ php artisan make:controller BookController --model=Book
```

This command create a `/app/Http/Controllers/BookController.php` file. Now open this file.

```php
<?php

namespace App\Http\Controllers;

use App\Models\Book;
use Illuminate\Http\Request;

class BookController extends Controller
{
    /**
     * Display a listing of the resource.
     */
    public function index()
    {
        //
```

```
    }

    /**
     * Show the form for creating a new resource.
     */
    public function create()
    {
        //
    }

    // ...........
    // ............
    // ...........
    // ...........
}
```

Now we are step by step to create **route**, implement BookController method and corresponding **views**.

/routes/web.php,

```php
// ---------------- Initial code --------------------
<?php

use Illuminate\Support\Facades\Route;

// comment out the older default route

// Route::get('/', function () {
//     return view('welcome');
// });

// ------------- create our route ----------------
Route::get("/books", [BookController::class, "index"])-
>name("books.index");
```

/app/Http/Controllers/BookController.php,

```php
    // ..................
    public function index()
    {
        // $books = Book::all();
        $books = Book::paginate(10);

        return view("books.index", ["books"=> $books]);
    }
    // .................
```

Now create a folder inside /resources/views/ name books and inside books folder create a file name, index.blade.php.

```
$ cd resources/views/
$ mkdir books
$ cd books
$ touch index.blade.php
```

Now open index.blade.php,

Add Bootstrap starter code and a Bootstrap table.

```html
<!doctype html>
<html lang="en">

<head>
    <meta charset="utf-8">
    <meta name="viewport" content="width=device-width, initial-scale=1">
    <link
href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/css/bootstrap.min.c
ss" rel="stylesheet"
        integrity="sha384-
EVSTQN3/azprG1Anm3QDgpJLIm9Nao0Yz1ztcQTwFspd3yD65VohhpuuCOmLASjC"
crossorigin="anonymous">

    <title>Bookstore</title>
</head>

<body>
    <div class="container">
        <h1>Bookstore - Information</h1>

        <table class="table">
            <thead>
                <tr>
                    <th scope="col">ID</th>
                    <th scope="col">Title</th>
                    <th scope="col">Author</th>
                    <th scope="col">ISBN</th>
                    <th scope="col">Price</th>
                    <th scope="col">Availabe</th>
                </tr>
            </thead>
            <tbody>
                @@foreach ($books as $book)
                <tr>
                    <th scope="row"> {{ $book.id }} </th>
                    <td> {{ $book.title  }} </td>
                    <td> {{ $book.author  }} </td>
                    <td> {{ $book.isbn  }} </td>
```

```
                    <td> {{ $book.price  }} </td>
                    <td> {{ $book.available  }} </td>
                </tr>
                @endforeach
            </tbody>
        </table>

    </div>

    <script
src="https://cdn.jsdelivr.net/npm/bootstrap@5.0.2/dist/js/bootstrap.bundle.
min.js"
        integrity="sha384-
MrcW6ZMFYlzcLA8Nl+NtUVF0sA7MsXsP1UyJoMp4YLEuNSfAP+JcXn/tWtIaxVXM"
crossorigin="anonymous">
    </script>
</body>

</html>
```

Now added pagination in view page, for that,

Little modify in `/resources/views/books/index.blade.php` file.... add little code below the `</table>` tag.

```
        </tbody>
        </table>

        {{ $books->links() }}  # new line...

    </div>
```

And modify inside the `/app/Providers/AppServiceProvider.php` file.

```php
<?php

namespace App\Providers;

use Illuminate\Pagination\Paginator;                    # new line
use Illuminate\Support\ServiceProvider;

class AppServiceProvider extends ServiceProvider
{
    /**
     * Register any application services.
     */
    public function register(): void
    {
        //
```

```
    }

    /**
     * Bootstrap any application services.
     */
    public function boot(): void
    {
        Paginator::useBootstrap();              # new line
    }
}
```

Bookstore - Information

ID	Title	Author	ISBN	Price	Availabe
1	Karlie Rolfson	Tyra Rau II	9782300028205	289	6
2	Alexandra Hyatt Sr.	Dr. Rachael Monahan	9785684166037	228	5
3	Linwood Cummings Jr.	Jovany Rempel	9798684541650	297	4
4	Lulu Mann II	Vicenta Cole	9785937675286	215	14
5	Shanie Quigley	Eino Simonis Sr.	9784833698849	153	14
6	Fae Kutch	Rod Sawayn	9795981871305	266	10
7	Miles Jenkins Sr.	Nia Jacobi	9794541550636	256	11
8	Johann Kuhlman	Mia Blanda	9795577607592	190	10
9	Felicita Heaney	Dr. Millie Lynch	9783137310938	141	15
10	Felicia Schimmel	Tiara Gulgowski	9782743648503	155	19

‹ 1 2 3 4 5 ›